

A numerical approach to the reach-avoid problem for Markov decision processes

Nikolaos Kariotoglou^{a,1}, Maryam Kamgarpour^a, Tyler H. Summers^b, John Lygeros^a

^a*Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich 8092 Switzerland*

^b*Control, Optimization, and Networks Lab, Department of Mechanical Engineering, The University of Texas at Dallas, Texas 75080, USA*

Abstract

An important problem in stochastic control is the so-called reach-avoid problem where one maximizes the probability of reaching a target set while avoiding unsafe subsets of the state-space. We develop a computational method for the finite horizon reach-avoid problem for discrete-time Markov decision processes. Our approach is based on deriving an infinite dimensional linear program whose solution is equivalent to the value function of the reach-avoid problem. We then propose a tractable approximation to the infinite linear program by projecting the value function onto the span of a finite number of basis functions and using randomized sampling to relax the infinite constraints. We illustrate the applicability of the method on a large class of Markov decision processes modeled by Gaussian mixtures and develop benchmark control and robotic problems to explore the computational tractability and accuracy of the proposed approach. To the best of our knowledge, this is the first time that problems up to six state variables and two inputs have been addressed using the reach-avoid formulation. Our results demonstrate that the approximation scheme has considerable computational advantages compared to standard space gridding-based methods.

Keywords: Reachability, approximate dynamic programming, Markov decision processes, stochastic control.

1. Introduction

A wide range of controlled physical systems can be modeled using the framework of Markov decision processes (MDPs) [1, 2]. The most common objective is to compute control policies for the MDP that maximize an expected reward, or minimize an expected cost over a finite or infinite time horizon. In this work, we consider the stochastic reach-avoid problem for MDPs in which one maximizes the probability of reaching a target set while staying in a safe set [3]. Several different versions of the reach-avoid problem have been used in the literature to model safety and performance specifications for dynamical systems with hybrid dynamics such as room temperature regulation [3], aircraft conflict resolution [4], power system stability [5], camera networks [6] and building evacuation strategies under randomly evolving hazards [7].

State-of-the-art solution methods express the stochastic reach-avoid problem as a dynamic programming (DP) recursion and then approximate it on a finite grid placed on the state and control spaces. Gridding techniques are theoretically attractive since they can provide explicit error bounds for the approximation of the value function under general Lipschitz continuity assumptions [8, 9, 10]. In practice, the complexity of gridding based techniques suffers from the infamous Curse of Dimensionality and thus, is only applicable to relatively low dimensional problems. For general stochastic reach-avoid problems, the sum of state and

Email addresses: karioto@control.ee.ethz.ch (Nikolaos Kariotoglou), mkamgar@control.ee.ethz.ch (Maryam Kamgarpour), tyler.summers@utdallas.edu (Tyler H. Summers), lygeros@control.ee.ethz.ch (John Lygeros)

¹The work of N. Kariotoglou was supported by the Swiss National Science Foundation under grant number 200021.137876.

control space dimensions that can be addressed with existing tools is not more than five. An important problem is therefore to explore alternative approximation techniques to push this limit further.

Several researchers have developed approximate dynamic programming (ADP) techniques for various classes of stochastic control problems [11, 12]. Most of the existing work has focused on problems where the state and control spaces are finite but too large to directly solve DP recursions. Our work is motivated by the technique discussed in [13] where the authors develop an approximation method based on Linear Programming (LP) for finite state and control spaces. Although the LP approach has been extensively used to approximate standard MDP problems with additive stage costs, it has not yet been considered for stochastic reach-avoid problems where the stage cost is sum-multiplicative. In general, LP approaches to ADP are desirable since several commercially available software packages can handle LP problems with large numbers of decision variables and constraints.

We develop a method to approximate stochastic reach-avoid problems over uncountable state and control spaces using linear programming. Our contributions can be summarized in the following points:

- We show the equivalence of a DP recursion that solves the reach-avoid problem and an infinite dimensional LP formulated over the space of Borel measurable functions defined on the MDP state space.
- We show that restricting the infinite dimensional space to a finite dimensional subspace spanned by a collection of basis functions is equivalent to the projection of the reach-avoid value functions onto the intersection of the span with the feasible region of the infinite LP. We then prove that recursively projecting the approximate value functions results in computing an upper bounding function for the reach-avoid probability over the MDP state space.
- We present a novel algorithm based on scenario sampling to approximate the value function of reach-avoid problems resulting in solving finite dimensional LP problems.
- We present a numerical implementation of our method using finite dimensional subspaces spanned by Gaussian radial basis functions (GRBFs) and demonstrate that this choice allows the analytic computation of integrals for a wide class of safe and target sets and a wide class of MDP transition kernels.
- We provide a series of numerical examples where we approximate the solution to reach-avoid dynamic recursions that are intractable using state-space gridding. In particular, we formulate two types of reach-avoid problems for which heuristic solutions can be efficiently computed using existing control methods and confirm that our approximations are accurate by comparing their performance. We also formulate a more complex reach-avoid problem inspired by car racing where constructing a benchmark method is not straightforward but our method is still applicable.

The rest of the paper is organized as follows: In Section 2 we present the basics of stochastic reach-avoid problems for MDPs with uncountable state and control spaces and formulate an infinite dimensional LP for which the reach-avoid value function is an optimal solution. In Section 3 we derive a numerical approach to approximate the solution to the infinite LP through restricting the decision space using basis functions and using randomized algorithms to sample the infinite constraints. Section 4 restricts the form of basis elements to GRBFs and analyzes some of their approximation and numerical computation benefits. We use the developed methodology in Section 5 to approximate three reach-avoid problems of varying structure and complexity. In the concluding section of the paper we summarize the main results and outline potential future work.

2. Stochastic Reach-Avoid Problem

Our goal in this section is to present the reach-avoid problem for MDPs and develop an infinite dimensional linear program that can characterize its solution. We consider a discrete-time controlled stochastic process $x_{t+1} \sim Q(dx|x_t, u_t)$, $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$ with a transition kernel $Q : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \rightarrow [0, 1]$ where

$\mathcal{B}(\mathcal{X})$ denotes the Borel σ -algebra of \mathcal{X} . Given a state control pair $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$, $Q(A|x_t, u_t)$ measures the probability of x_{t+1} falling in a set $A \in \mathcal{B}(\mathcal{X})$. The transition kernel Q is a Borel-measurable stochastic kernel, that is, $Q(A|\cdot)$ is a Borel-measurable function on $\mathcal{X} \times \mathcal{U}$ for each $A \in \mathcal{B}(\mathcal{X})$ and $Q(\cdot|x, u)$ is a probability measure on \mathcal{X} for each (x, u) . For the rest of the paper all measurability conditions refer to Borel measurability. We allow the state space \mathcal{X} to be any subset of \mathbb{R}^n and assume that the control space $\mathcal{U} \subseteq \mathbb{R}^m$ is compact; extensions to hybrid state and input spaces where some state or inputs are finite valued are possible [3]. We consider a safe set $K' \in \mathcal{B}(\mathcal{X})$ and a target set $K \subseteq K'$. We define an admissible T -step control policy to be a sequence of measurable functions $\mu = \{\mu_0, \dots, \mu_{T-1}\}$ where $\mu_i : \mathcal{X} \rightarrow \mathcal{U}$ for each $i \in \{0, \dots, T-1\}$. The reach-avoid problem over a finite time horizon T is to find an admissible T -step control policy that maximizes the probability of x_t reaching the set K at some time $t_K \leq T$ while staying in K' for all $t \leq t_K$. For any initial state x_0 we denote the reach-avoid probability associated with a given μ as: $r_{x_0}^\mu(K, K') = \mathbb{P}_{x_0}^\mu \{\exists j \in [0, T] : x_j \in K \wedge \forall i \in [0, j-1], x_i \in K' \setminus K\}$ and operate under the assumption that $[0, -1] = \emptyset$ which implies that the requirement on i is automatically satisfied when $x_0 \in K$.

2.1. Dynamic programming approach

In [14], $r_{x_0}^\mu(K, K')$ is shown to be equivalent to the following sum multiplicative cost function:

$$r_{x_0}^\mu(K, K') = \mathbb{E}_{x_0}^\mu \left[\sum_{j=0}^T \left(\prod_{i=0}^{j-1} \mathbb{1}_{K' \setminus K}(x_i) \right) \mathbb{1}_K(x_j) \right] \quad (1)$$

where $\prod_{i=k}^j (\cdot) = 1$ if $k > j$. The function $\mathbb{1}_A(x)$ denotes the indicator function of a set $A \in \mathcal{B}(\mathcal{X})$ with $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ otherwise. The sets K and K' can be time dependent or even stochastic [15] but for simplicity we assume here that they are constant. We denote the difference between the safe and target sets by $\bar{\mathcal{X}} := K' \setminus K$ to simplify the presentation of our results.

The solution to the reach-avoid problem is given by a dynamic recursion [14]. Define $V_k^* : \mathcal{X} \rightarrow [0, 1]$ for $k = T-1, \dots, 0$ by :

$$\begin{aligned} V_k^*(x) &= \sup_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x, u) \right\} \\ V_T^*(x) &= \mathbb{1}_K(x). \end{aligned} \quad (2)$$

The value of the above recursion at $k = 0$ is the supremum of (1) over all admissible policies, i.e. $V_0^*(x_0) = \sup_{\mu} r_{x_0}^\mu(K, K')$. In [14] the authors proved universal measurability of the value functions in (2). Here we impose a mild additional assumption on the continuity of the transition kernel and show that the value functions are measurable and the supremum at each step is attained.

Assumption 1. *For every $x \in \mathcal{X}$, $A \in \mathcal{B}(X)$ the mapping $u \mapsto Q(A|x, u)$ is continuous.*

Proposition 1. *Under Assumption 1, for every k the supremum in (2) is attained by a measurable function $\mu_k^* : \mathcal{X} \rightarrow \mathcal{U}$ and the resulting $V_k^* : \mathcal{X} \rightarrow [0, 1]$ is measurable.*

Proof. By induction. First, note that the indicator function $V_T^*(x) = \mathbb{1}_K(x)$ is measurable. Assuming that V_{k+1}^* is measurable we will show that V_k^* is also measurable. Define $F(x, u) = \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x, u)$. Due to continuity of the map $u \mapsto Q(A|x, u)$ by Assumption 1, the mapping $u \mapsto F(x, u)$ is continuous for every x (by [16, Fact 3.9]). Now, since \mathcal{U} is compact, by [17, Corollary 1], there exists a measurable function $\mu_k^*(x)$ that achieves the supremum. Furthermore, by [18, Proposition 7.29], the mapping $(x, u) \mapsto F(x, u)$ is measurable. It follows that $F(x, \mu_k^*(x))$ (and hence V_k^*) is measurable as it is the composition of measurable functions. We conclude by induction that at each time step k there exists a measurable function μ_k^* achieving the supremum and that V_k^* is measurable for every k . \square

Proposition 1 allows one to compute an optimal feedback policy at each stage k by solving,

$$\mu_k^*(x) = \arg \max_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x, u) \right\}. \quad (3)$$

The dynamic recursion in (2) implies that the functions $V_k^*(x)$ are defined on three disjoint regions of \mathcal{X} , namely

$$V_k^*(x) = \begin{cases} 1, & x \in K \\ \max_{u \in \mathcal{U}} \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x, u), & x \in \bar{\mathcal{X}} \\ 0, & x \in \mathcal{X} \setminus \bar{\mathcal{X}} \end{cases} \quad (4)$$

and it can be shown [14] that the value of each V_k^* on \mathcal{X} is restricted to $[0, 1]$. Hence, it suffices to compute V_k^* on $\bar{\mathcal{X}}$.

Proposition 2. *Assume that for every $A \in \mathcal{B}(\mathcal{X})$ the mapping $(x, u) \mapsto Q(A|x, u)$ is continuous (stronger than Assumption 1), then $V_k^*(x)$ is piecewise continuous on \mathcal{X} .*

Proof. From the continuity of $(x, u) \mapsto Q(A|x, u)$ we conclude that the mapping $(x, u) \mapsto F(x, u)$ is continuous (Fact 3.9 in [16]). From the Maximum Theorem [19], it follows that $F(x, u^*(x))$ and thus each $V_k^*(x)$, is continuous on $\bar{\mathcal{X}}$. By construction, each V_k^* is then piecewise continuous on \mathcal{X} . \square

2.2. Linear programming approach

We express the reach-avoid DP value function as a solution to an infinite dimensional LP, that we will later approximate using tools from function approximation and randomized convex optimization. Let $\mathcal{F} := \{f : \mathcal{X} \rightarrow \mathbb{R}, f \text{ is measurable}\}$. Under the conditions of Proposition 1, we introduce two operators defined for any measurable function $V \in \mathcal{F}$ to simplify the presentation of our results:

$$\begin{aligned} \mathcal{T}_u[V](x) &= \int_{\mathcal{X}} V(y) Q(dy|x, u) \\ \mathcal{T}[V](x) &= \max_{u \in \mathcal{U}} \mathcal{T}_u[V](x). \end{aligned} \quad (5)$$

The value function at each step $k \in \{0, \dots, T-1\}$ can be written as $V_k^*(x) = \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \mathcal{T}[V_{k+1}^*](x)$. Since $\mathbb{1}_K$ and $\mathbb{1}_{\bar{\mathcal{X}}}$ are known functions, one only needs to compute the value of $\mathcal{T}[V_{k+1}^*](x)$ on $\bar{\mathcal{X}}$ recursively for every $k \in \{0, \dots, T-1\}$ in order to construct each V_k^* using $\mathbb{1}_K$ and $\mathbb{1}_{\bar{\mathcal{X}}}$. In the following proposition, we show that for every $k \in \{0, \dots, T-1\}$ the solution to the recursive step in (2) can be constructed from the solution of an infinite dimensional LP using a relaxed version $V_k^*(x) \geq \mathcal{T}_u[V_{k+1}^*](x)$, $\forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}$ of the equation $V_k^*(x) = \mathcal{T}[V_{k+1}^*](x)$, $\forall x \in \bar{\mathcal{X}}$. For the rest of the paper let ν be a non-negative measure supported on $\bar{\mathcal{X}}$.

Proposition 3. *Suppose Assumption 1 holds. For $k \in \{0, \dots, T-1\}$, let V_{k+1}^* be the value function at step $k+1$ in (2). Define the following infinite dimensional linear program:*

$$\begin{aligned} J^* &:= \inf_{V(\cdot) \in \mathcal{F}} \int_{\bar{\mathcal{X}}} V(x) \nu(dx) \\ \text{subject to } & V(x) \geq \mathcal{T}_u[V_{k+1}^*](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \quad (6)$$

Then, V_k^ is a solution to (6) and any other solution to (6) is equal to V_k^* ν -almost everywhere on $\bar{\mathcal{X}}$.*

Proof. From Proposition 1, $V_k^* \in \mathcal{F}$ and is equal to the supremum over $u \in \mathcal{U}$ of the right hand side of the constraint in (6). Hence, for any feasible $V \in \mathcal{F}$, we have that $V(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$ and thus $\int_{\bar{\mathcal{X}}} V(x) \nu(dx) \geq \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$ which implies $J^* \geq \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$. On the other hand, $J^* \leq \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$ since it is the least cost among the set of feasible measurable functions. Putting these together we have that

$J^* = \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$ and V_k^* is an optimal solution. We now show that any other solution to (6) is equal to V_k^* ν -almost everywhere on $\bar{\mathcal{X}}$. Assume that there exists a function V^* optimal for (6) that is strictly greater than V_k^* on a set $A_m \in \mathcal{B}(\mathcal{X})$ of non-zero ν -measure. Since V^* and V_k^* are both optimal, we have that $\int_{\bar{\mathcal{X}}} V^*(x) \nu(dx) = \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx) = J^*$. We can then reduce V^* to the value of V_k^* on A_m while keeping it measurable, reducing the value of $\int_{\bar{\mathcal{X}}} V^*(x) \nu(dx)$ below J^* , contradicting the fact that V^* was optimal. We can then conclude that any solution to (6) has to be equal to V_k^* ν -almost everywhere on $\bar{\mathcal{X}}$. \square

Note that we can replace the infimum in (6) by a minimum since the infimum is attained by the value function. Consider the following semi-norm on \mathcal{F} induced by the non-negative measure ν :

Definition 1 (ν -norm). $\|V\|_{1,\nu} := \int_{\bar{\mathcal{X}}} |V(x)| \nu(dx)$.

Any feasible function in (6) is a point-wise upper bound on the value function V_k^* and Proposition 3 implies that for any solution V^* to the infinite dimensional LP in (6) we have $\|V^* - V_k^*\|_{1,\nu} = 0$. The results of Proposition 3 hold for any non-negative measure but choosing ν poorly affects the quality of a solution to (6) as discussed in [13]. In theory, we could start at $k = T - 1$ and recursively solve problem (6) to construct the required value function V_0^* , up to a set of ν -measure zero. However, computing a solution to infinite dimensional LP problems in the form of (6) is generally intractable [20, 21] unless the set \mathcal{F} has a finite dimensional representation and the infinite constraints over $\bar{\mathcal{X}} \times \mathcal{U}$ can be reformulated or evaluated efficiently.

3. Approximation with a finite linear program

The goal of this section is to develop a method for approximating the intractable infinite dimensional LP by a finite LP through two steps. First, we replace the infinite dimensional function space \mathcal{F} with a finite dimensional one and discuss the effects of this restriction. Second, we replace the infinite constraints enforced on $\bar{\mathcal{X}} \times \mathcal{U}$ with a finite number of constraints, and provide probabilistic feasibility guarantees for the rest.

3.1. Restriction to a finite dimensional function class

Let \mathcal{F}^M be a finite dimensional subspace of \mathcal{F} spanned by M measurable basis elements denoted by $\{\phi_i\}_{i=1}^M$. For a fixed function $f \in \mathcal{F}$, consider the following semi-infinite LP defined over functions $V \in \mathcal{F}^M$ expressed using the basis $\{\phi_i\}_{i=1}^M$ as $V(x) = \sum_{i=1}^M w_i \phi_i(x)$ for some $w \in \mathbb{R}^M$,

$$\begin{aligned} \min_{w_1, \dots, w_M} \quad & \sum_{i=1}^M w_i \int_{\bar{\mathcal{X}}} \phi_i(x) \nu(dx) \\ \text{subject to} \quad & \sum_{i=1}^M w_i \phi_i(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \\ & w \in \mathbb{R}^M. \end{aligned} \tag{7}$$

The following proposition relates the solution of (7) to the corresponding infinite dimensional LP defined over the whole space \mathcal{F} .

Lemma 1. *Suppose Assumption 1 holds. Fix a function $f \in \mathcal{F}$ and let V^* be a solution to the following optimization problem:*

$$\begin{aligned} \min_{V(\cdot) \in \mathcal{F}} \quad & \int_{\bar{\mathcal{X}}} V(x) \nu(dx) \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \tag{8}$$

Denote by $\hat{V}(x) = \sum_{i=1}^M \hat{w}_i \phi_i(x)$ the function constructed using the optimal solution \hat{w} of (7) for the same function f . Then $\hat{V}(x) \geq V^*(x)$ everywhere on $\bar{\mathcal{X}}$ and \hat{V} is a solution to the following optimization problem:

$$\begin{aligned} & \min_{V(\cdot) \in \mathcal{F}^M} \|V - V^*\|_{1,\nu} \\ & \text{subject to } V(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \quad (9)$$

Proof. Any feasible V in (8) upper bounds V^* on $\bar{\mathcal{X}}$, i.e. $V(x) \geq V^*(x)$ for all $x \in \bar{\mathcal{X}}$ and in particular, since $\hat{V} \in \mathcal{F}$, $\hat{V}(x) \geq V^*(x)$ for all $x \in \bar{\mathcal{X}}$. To show that \hat{V} is a solution to (9) consider

$$\|V - V^*\|_{1,\nu} = \int_{\bar{\mathcal{X}}} V(x) \nu(dx) - \int_{\bar{\mathcal{X}}} V^*(x) \nu(dx) = \int_{\bar{\mathcal{X}}} V(x) \nu(dx) - C$$

where C is equal to the integral of V^* on $\bar{\mathcal{X}}$ with respect to ν and is a constant since V^* is fixed. Since any function in $V \in \mathcal{F}^M$ can be written as $V(x) = \sum_{i=1}^M w_i \phi_i(x)$, problem (7) is equivalent to (9) for the same $f \in \mathcal{F}$. \square

The semi-infinite optimization problem in (7) can be used to recursively construct an approximation \hat{V}_0 to V_0^* on $\bar{\mathcal{X}}$ where each approximation step \hat{V}_k satisfies Lemma 1.

Proposition 4. For every $k \in \{0, \dots, T-1\}$ let \mathcal{F}^{M_k} denote the span of a fixed set of M_k measurable basis elements denoted by $\{\phi_i^k\}_{i=1}^{M_k}$. Start with the known value function V_T^* and recursively construct $\hat{V}_{T-1}(x), \dots, \hat{V}_0(x)$ where \hat{w}_i^k and $\hat{V}_k(x) = \sum_{i=1}^{M_k} \hat{w}_i^k \phi_i^k(x)$ are obtained by substituting $f = \hat{V}_{k+1}$ in (7). Each function \hat{V}_k is then a solution to the corresponding optimization problem:

$$\begin{aligned} & \min_{V(\cdot) \in \mathcal{F}^{M_k}} \|V - V_k^*\|_{1,\nu} \\ & \text{subject to } V(x) \geq \mathcal{T}_u[\hat{V}_{k+1}](x) \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \end{aligned} \quad (10)$$

and $\hat{V}_k(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$.

Proof. The statement is a recursive application of Lemma 1. For $k = T-1$, the constraint in (10) implies that $\hat{V}_{T-1}(x) \geq \mathcal{T}[V_T^*](x) = V_{T-1}^*(x)$. Applying \mathcal{T} on both sides we have by the monotonicity of \mathcal{T} (see [14]) that $\mathcal{T}[\hat{V}_{T-1}](x) \geq \mathcal{T}[V_{T-1}^*](x) = V_{T-2}^*(x)$. By the constraints in (10) at $k = T-2$ we have that $\hat{V}_{T-2}(x) \geq \mathcal{T}[\hat{V}_{T-1}](x)$ which implies that $\hat{V}_{T-2}(x) \geq V_{T-2}^*(x)$. Repeating, we arrive to the result $\hat{V}_0(x) \geq V_0^*(x)$. Using the fact that $\hat{V}_k(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$ we conclude in the same way as in Lemma 1 that $\hat{V}_k(x)$ minimizes the ν -norm distance to V_k^* while respecting the corresponding constraint. \square

3.2. Restriction to a finite number of constraints

Finite dimensional problems with infinite constraints (similar to the ones appearing in the recursive process of Proposition 4) are called semi-infinite or robust optimization problems. These are difficult to solve unless specific structure is imposed on $\bar{\mathcal{X}} \times \mathcal{U}$ and the basis functions ([22, 23, 24]). An alternative approach is to select a finite set of points from $\bar{\mathcal{X}} \times \mathcal{U}$ and impose the constraints only on those points. Problems in the form of (7) are then reduced to finite LP problems and can be solved to optimality using commercially available solvers. We will use the scenario approach [25] to quantify the feasibility properties of solutions constructed using sampled data. For a discussion on the related performance properties, see [26, 27]. Let $S := \{(x^i, u^i)\}_{i=1}^N$ denote a set of $N \in \mathbb{N}$ elements from $\bar{\mathcal{X}} \times \mathcal{U}$ and for a fixed function $f \in \mathcal{F}$, consider the following finite LP defined over functions $V \in \mathcal{F}^M$ expressed using the basis $\{\phi_i\}_{i=1}^M$ as

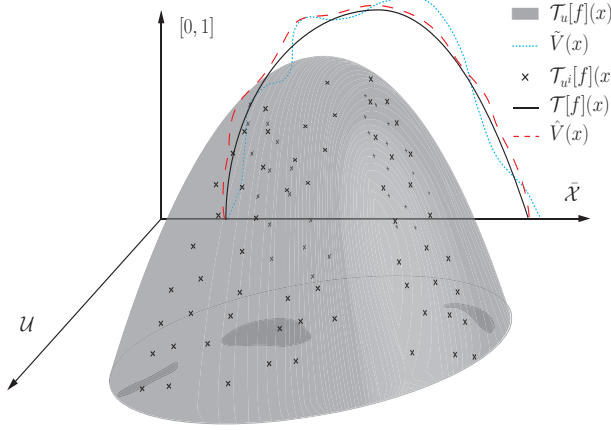


Figure 1: Illustrative example of an approximation after sampling depicted on $\bar{\mathcal{X}} \times \mathcal{U}$.

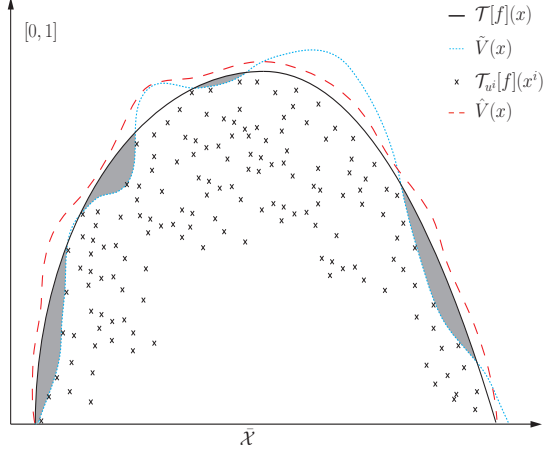


Figure 2: Illustrative example of an approximation after sampling, projected on $\bar{\mathcal{X}}$.

$$V(x) = \sum_{i=1}^M w_i \phi_i(x) \text{ for some } w \in \mathbb{R}^M$$

$$\begin{aligned} \min_{w_1, \dots, w_M} \quad & \sum_{i=1}^M w_i \int_{\bar{\mathcal{X}}} \phi_i(x) \nu(dx) \\ \text{subject to} \quad & \sum_{i=1}^M w_i \phi_i(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in S \\ & w \in \mathbb{R}^M. \end{aligned} \tag{11}$$

Since the objective and constraint functions are linear in the decision variables, we use the scenario sampling theorem from [25] and generalize the feasibility properties of a solution to (11) with respect to a solution to (7), as a function of the set S . For the problem at hand, the following statement summarizes the probabilistic feasibility guarantees of a solution constructed in this way.

Theorem 1. Assume that for any function $f \in \mathcal{F}$ and any set S , the feasible region of (11) is non-empty and the optimizer is unique. Fix a function $f \in \mathcal{F}$ and choose a collection of functions $\{\phi_i\}_{i=1}^M$ and required violation and confidence levels $\varepsilon, \beta \in (0, 1)$. Construct a set S by drawing N independent identically distributed sample points from $\bar{\mathcal{X}} \times \mathcal{U}$ according to some probability measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ where N is chosen as

$$N \geq S(\varepsilon, \beta, M) = \min \left\{ N \in \mathbb{N} \mid \sum_{i=0}^{M-1} \binom{N}{i} \varepsilon^i (1 - \varepsilon)^{N-i} \leq \beta \right\}.$$

Then the function $\tilde{V}(x) = \sum_{i=1}^M \tilde{w}_i \phi_i(x)$ constructed using the optimal solution \tilde{w} to (11) satisfies

$$\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}} \left[\tilde{V}(x) < \mathcal{T}_u[f](x) \right] \leq \varepsilon \tag{12}$$

with confidence $1 - \beta$, measured with respect to $\mathbb{P}_{\{\bar{\mathcal{X}} \times \mathcal{U}\}^N}$, the measure constructed by taking N products of $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$.

The probability statement in (12) can be interpreted as follows: The approximate solution function $\tilde{V}(x) = \sum_{i=1}^M \tilde{w}_i \phi_i(x)$ violates the constraint $\mathcal{T}_u[f](x)$ on a set of at most ε measure, with respect to $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$. In contrast to the solution in (7), the constructed value functions are upper bounds on $\bar{\mathcal{X}}$ in probability. To further clarify the statement of Theorem 1, consider the illustration in Figures 1 and 2. Assuming that $\bar{\mathcal{X}}$

and \mathcal{U} are one dimensional and for a hypothetical function f , we have plotted the value of the constraint $\mathcal{T}_u[f](x^i)$ (denoted by crosses) for a collection of sampled points (x^i, u^i) . The dotted lines correspond to the solution of (11) for the particular f , denoted by $\tilde{V}(x)$. The dashed lines correspond to the solution of (7) for the particular f , denoted by $\hat{V}(x)$, while the solid lines correspond to $\mathcal{T}[f](x)$. As stated in Theorem 1, $\tilde{V}(x)$ is greater than or equal to $\mathcal{T}_u[f](x)$ for most pairs (x, u) . $\tilde{V}(x)$ could exceed or stay below $\mathcal{T}[f](x)$, as long as the measure of pairs (x, u) where it is below is smaller than ε_k . In Figure 1, we have plotted the sets where $\tilde{V}(x) < \mathcal{T}_u[f](x)$ on $\tilde{\mathcal{X}} \times \mathcal{U}$. In Figure 2, the shaded areas only indicate that there exist pairs (x, u) such that $\tilde{V}(x) < \mathcal{T}_u[f](x)$.

Proposition 5. *For every $k \in \{0, \dots, T-1\}$ let \mathcal{F}^{M_k} denote the span of a fixed set of $M_k \in \mathbb{N}_+$ basis elements $\{\phi_i^k\}_{i=1}^{M_k}$ and choose levels $\{\varepsilon_i, \beta_i \in (0, 1)\}_{i=1}^{M_k}$. Start with the known value function $V_T^* \in \mathcal{F}$ and recursively construct $\tilde{V}_k \in \mathcal{F}^{M_k}$ by solving (11) using $f = \tilde{V}_{k+1}$, $\{\phi_i\}_{i=1}^M = \{\phi_i^k\}_{i=1}^{M_k}$, $S = S_k$ where S_k is constructed using $\varepsilon = \varepsilon_k$, $\beta = \beta_k$ and $M = M_k$ in Theorem 1. Denote by N_k the number of elements in S_k . Each function \tilde{V}_k then satisfies:*

$$\mathbb{P}_{\tilde{\mathcal{X}} \times \mathcal{U}} \left[\tilde{V}_k(x) < \mathcal{T}_u[\tilde{V}_{k+1}](x) \right] \leq \varepsilon_k \quad (13)$$

with confidence $1 - \beta_k$, measured with respect to $\mathbb{P}_{\{\tilde{\mathcal{X}} \times \mathcal{U}\}^{N_k}}$, the measure constructed by taking N_k products of $\mathbb{P}_{\tilde{\mathcal{X}} \times \mathcal{U}}$.

Since at every step k , the basis elements $\{\phi_i^k\}_{i=1}^{M_k}$, the function $\tilde{V}_{k+1} \in \mathcal{F}$ and the levels $\{\varepsilon_i, \beta_i \in (0, 1)\}_{i=1}^{M_k}$ are fixed, Proposition 5 is simply a recursive application of Theorem 1 for $k \in \{0, \dots, T-1\}$. In the following section we discuss a particular choice for the basis sets $\{\phi_i^k\}_{i=1}^{M_k}$ that provides a number of computational advantages in the context of reach-avoid problems.

4. Implementation with radial basis functions

We consider a particular type of radial basis functions and apply the presented methodology to a powerful class of MDPs modeled by Gaussian mixture kernels [28]. We show that using this class of basis functions, the constraint in (11) can be computed analytically for each element in the sample set, reducing the computation time required to solve the resulting linear program.

4.1. Basis function choice

We consider basis function sets comprising parametrized Gaussian radial basis functions (GRBFs). Our choice is motivated by the strong approximation capabilities of such functions, discussed in [29, 30, 31, 32], as well as the fact that for certain types of sets one can compute integrals over GRBFs analytically. Consider the following assumption:

Assumption 2. *We impose the following restrictions on the structure of the considered reach-avoid problems:*

1. *The stochastic kernel Q can be written as a Gaussian mixture kernel $\sum_{j=1}^J \alpha_j \mathcal{N}(\mu_j, \Sigma_j)$ with known diagonal covariance matrices Σ_j , means μ_j and weights α_j such that $\sum_{j=1}^J \alpha_j = 1$ for a finite $J \in \mathbb{N}_+$.*
2. *The target and safe sets K and K' can be written as finite unions of disjoint hyper-rectangle sets, i.e. $K = \bigcup_{p=1}^P K_p = \bigcup_{p=1}^P (\times_{l=1}^n [a_l^p, b_l^p])$ and $K' = \bigcup_{m=1}^M K'_m = \bigcup_{m=1}^M (\times_{l=1}^n [c_l^m, d_l^m])$ for some finite $P, M \in \mathbb{N}_+$ with $n = \dim(\mathcal{X})$ and $a^p, b^p, c^m, d^m \in \mathbb{R}^n$ for every p and m .*

Assumption 2 applies to a wide range of reach-avoid problems. For example the kernel of general non-linear systems subject to additive Gaussian mixture noise can be written as a GRBF. Moreover, whenever the safe and target sets cannot be written as unions of disjoint hyper-rectangles, one can approximate them as such to arbitrary accuracy [33]. Since the state-relevance measure ν is a design choice, it is generally possible to select one that can be written as a product measure; if this is not the case, our results extend to measures ν that can be approximated by Gaussian mixtures.

For each $k \in \{0, \dots, T-1\}$, \mathcal{F}^{M_k} denotes the span of a set of M_k GRBFs $\{\phi_i^k\}_{i=1}^{M_k}$ where for each $i \in \{1, \dots, M_k\}$, ϕ_i^k is a mapping from \mathbb{R}^n to \mathbb{R} that has the form:

$$\phi_i^k(x) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi s_l^{i,k}}} \exp\left(-\frac{1}{2} \frac{(x_l - c_l^{i,k})^2}{s_l^{i,k}}\right) \quad (14)$$

where $x, c^{i,k}, s^{i,k} \in \mathbb{R}^n$ and all $c^{i,k}, s^{i,k} \in \mathbb{R}$. Note that the expression in (14) is equivalent to a multi-variate Gaussian function with diagonal covariance matrix. The candidate approximate value function at stage $k \in \{0, \dots, T-1\}$ is constructed by taking a weighted sum $\sum_{i=1}^{M_k} w_i^k \phi_i^k(x)$ of the corresponding basis elements. For every $k \in \{0, \dots, T-1\}$, we fix the centers and variances of the GRBF elements $\{\phi_i^k\}_{i=1}^{M_k}$ before initializing the recursive process described in Proposition 5 to optimize over their scalar weights. We sample centers uniformly from the set $\bar{\mathcal{X}}$ and variances from a bounded set that varies according to problem data and constitutes a design choice (see Section 5).

A useful property of GRBFs is that the product of two GRBFs is a GRBF with known center and variance [29, Section 2]. Consider two functions $f^1, f^2 \in \mathcal{F}^{M_k}$ such that $f^1(x) = \sum_{i=1}^{M_k} w_i^1 \phi_i^k(x)$ and $f^2(x) = \sum_{j=1}^{M_k} w_j^2 \phi_j^k(x)$. We then have that $f(x) = f^1(x)f^2(x) = \sum_{i=1}^{M_k} \sum_{j=1}^{M_k} w_{i,j} \tilde{\phi}_{i,j}^k(x)$ where $w_{i,j} = w_i^1 w_j^2$ and $\tilde{\phi}_{i,j}^k$ is equal to

$$\tilde{\phi}_{i,j}^k(x) = \prod_{l=1}^n \frac{\gamma_l^{i,j,k}}{\sqrt{2\pi s_l^{i,j,k}}} \exp\left(-\frac{1}{2} \frac{(x_l - c_l^{i,j,k})^2}{s_l^{i,j,k}}\right)$$

with

$$c_l^{i,j,k} = \frac{c_l^{i,k} s_l^{j,k} + c_l^{j,k} s_l^{i,k}}{s_l^{i,k} + s_l^{j,k}}, \quad s_l^{i,j,k} = \sqrt{\frac{s_l^{i,k} s_l^{j,k}}{s_l^{i,k} + s_l^{j,k}}}, \quad \gamma_l^{i,j,k} = \frac{1}{\sqrt{2\pi(s_l^{i,k} + s_l^{j,k})}} \exp\left(-\frac{c_l^{i,k} - c_l^{j,k}}{2(s_l^{i,k} + s_l^{j,k})}\right).$$

Under the setup of Assumption 2, integrals of GRBFs over K and K' decompose into one dimensional integrals of Gaussian functions which can be computationally useful in the method of Proposition 5 that requires evaluating the constraint in (11) over samples from $\bar{\mathcal{X}} \times \mathcal{U}$. In particular, let $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$ denote the approximate value function at time k and $A = \bigcup_{d=1}^D A_d$ with each $A_d = \{[a_1^d, b_1^d] \times \dots \times [a_n^d, b_n^d]\}$, a union of hyper-rectangle sets for some $D \in \mathbb{N}_+$. The integral of \tilde{V}_k over A can then be written as

$$\begin{aligned} \int_A \tilde{V}_k(x) \nu(dx) &= \sum_{d=1}^D \int_{A_d} \tilde{V}_k(x) \nu(dx) \\ &= \sum_{d=1}^D \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{A_d} \phi_i^k(x) \nu(dx) = \sum_{d=1}^D \sum_{i=1}^{M_k} \tilde{w}_i^k \prod_{l=1}^n \int_{a_l^d}^{b_l^d} \frac{1}{\sqrt{2\pi s_l^{i,k}}} \exp\left(-\frac{1}{2} \frac{(x_l - c_l^{i,k})^2}{s_l^{i,k}}\right) \nu_l(dx_l) \\ &= \sum_{d=1}^D \sum_{i=1}^{M_k} \tilde{w}_i^k \prod_{l=1}^n -\frac{1}{2} \operatorname{erf}\left(\frac{x_l - c_l^{i,k} - b_l^d}{\sqrt{2s_l^{i,k}}}\right) + \frac{1}{2} \operatorname{erf}\left(\frac{x_l - c_l^{i,k} - a_l^d}{\sqrt{2s_l^{i,k}}}\right) \end{aligned} \quad (15)$$

where erf denotes the error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$. Note that we choose the state-relevance measure ν to be a product measure, i.e. $\nu(dx) = \prod_{l=1}^n \nu_l(dx_l)$ in order to carry out this

computation. The integral of \tilde{V}_k over \mathcal{X} can then be written as

$$\begin{aligned} \int_{\mathcal{X}} \tilde{V}_k(y) Q(dy|x, u) &= \sum_{m=1}^M \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{K'_m} \phi_i^k(y) Q(dy|x, u) - \sum_{p=1}^P \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{K_p} \phi_i^k(y) Q(dy|x, u) \\ &\quad + \sum_{p=1}^P \int_{K_p} Q(dy|x, u) \end{aligned} \quad (16)$$

and since Q can be written as a GRBF, every product in (16) is a product of GRBFs and the integrals over K_p and K'_m can be computed using (15).

4.2. Recursive value function approximation and policy computation

For a given horizon T , space $\bar{\mathcal{X}} \times \mathcal{U}$, transition kernel Q and target and safe sets K and K' , the approximate value functions corresponding to (2) are computed using Proposition 5. The method relies on recursively applying Theorem 1 which depends on several parameters that constitute design choices. The first and most important choice is the number of basis elements $\{M_k\}_{k=0}^{T-1}$ used at every step of the approximation that, as seen in (11) and the sample bound in Theorem 1, affects both the number of decision variables and constraints. The samples are generated according to a chosen measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ and, apart from $\{M_k\}_{k=0}^{T-1}$, their number is also affected by the choice of violation and confidence levels $\{\varepsilon_i\}_{i=0}^{T-1}$, $\{1 - \beta_i\}_{i=0}^{T-1}$. Typically, violation and confidence levels are chosen to be close to 0 and 1 respectively to enhance the feasibility guarantees of Theorem 1 (measured using $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$) at the expense of more constraints in (11). The quality of the resulting approximation depends on the choice of the center and variance parameters of each basis set $\{\phi_i^k\}_{i=1}^{M_k}$ as well as the state-relevance measure ν , given as a product of measures. At each recursive step $k \in \{0, \dots, T-1\}$ we construct \mathcal{F}^{M_k} by sampling M_k centers from $\bar{\mathcal{X}}$ and variances from a bounded set, according to a chosen probability measure; our method is still applicable if centers and variances are not sampled but fixed in another way. We then sample points from $\bar{\mathcal{X}} \times \mathcal{U}$ according to $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ and the sample bound of Theorem 1 and evaluate the constraints in (11) using (15) and (16). The approximate value function at step k is constructed using the optimal solution of the resulting finite LP and the process is repeated until $k = 0$. The steps of the proposed method are summarized in Algorithm 1.

Using the approximate value functions one can compute an approximate control policy for any $x \in \bar{\mathcal{X}}$ and $k \in \{0, \dots, T-1\}$ by solving:

$$\begin{aligned} \tilde{\mu}_k(x) &= \arg \max_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} \tilde{V}_{k+1}(y) Q(dy|x, u) \right\} \\ &= \arg \max_{u \in \mathcal{U}} \int_{\mathcal{X}} \tilde{V}_{k+1}(y) Q(dy|x, u). \end{aligned} \quad (17)$$

Although the optimization problem in (17) is non-convex, standard gradient based algorithms can be employed to obtain a local solution. In particular, the cost function is by construction smooth with respect to u for a fixed $x \in \bar{\mathcal{X}}$ and the gradient and Hessian information can be analytically obtained using the erf function. Moreover, the decision space \mathcal{U} is typically low dimensional (in most mechanical systems for example $\dim \mathcal{U} \leq \dim \mathcal{X}$) and mature software is available [34] to compute locally optimal solutions. The process of calculating a control input at time k for a fixed state x_k is summarized in Algorithm 2. An alternative approach would be to use randomized techniques similar to the approaches in [25, 35, 36, 37].

5. Numerical Case Studies

A challenge in evaluating approximate dynamic programming methods is the construction of benchmark problems since exact solutions are not available. For this reason, we consider benchmark problems based on some of the most powerful and scalable existing tools to explore the performance of our method. We analyze the numerical performance of the proposed method on three reach-avoid problems that differ in structure

Algorithm 1 Approximate value function

Input Data:

- State and control space $\bar{\mathcal{X}} \times \mathcal{U}$.
- Reach-avoid time horizon T .
- Centers and variances of the MDP kernel Q .
- Sets K and K' written as unions of disjoint hyper-rectangles.

Design parameters:

- Number of basis elements $\{M_k\}_{k=0}^{T-1}$.
- Violation and confidence levels $\{\varepsilon_i\}_{i=0}^{T-1}, \{1 - \beta_i\}_{i=0}^{T-1}$.
- Probability measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ used in Theorem 1.
- Probability measure of centers and variances for the basis functions $\{\phi_i^k\}_{i=1}^{M_k}$, supported on $\bar{\mathcal{X}}$ and a bounded set respectively.
- State-relevance measure ν decomposed as a product measure.

Initialize $\tilde{V}_T(x) \leftarrow \mathbb{1}_K(x)$.

for $k = T - 1$ **to** $k = 0$ **do**

Construct \mathcal{F}^{M_k} by sampling M_k centers $\{c_i\}_{i=1}^{M_k}$ and variances $\{s_i\}_{i=1}^{M_k}$ according to the chosen probability measures.

Sample $S(\varepsilon_k, \beta_k, M_k)$ pairs (x^s, u^s) from $\bar{\mathcal{X}} \times \mathcal{U}$ using the measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$.

for all (x^s, u^s) **do**

Evaluate $\mathcal{T}_{u^s}[\tilde{V}_{k+1}](x^s)$ using (15) and (16).

end for

Solve the finite LP in (11) to obtain $\tilde{w}^k = (\tilde{w}_1^k, \dots, \tilde{w}_{M_k}^k)$.

Set the approximated value function on $\bar{\mathcal{X}}$ to $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$.

end for

and complexity. The first two examples correspond to regulation problems with different constraint types and are used to demonstrate the accuracy of the method as the state and control space dimensions grow. The final example demonstrates the potential of the proposed approach by addressing a reach-avoid problem for a highly non-linear system with six state variables and two control inputs. To compute the ADP reach-avoid controllers for all problems, we have used the method described in Algorithm 1 without exploiting any additional structure in the systems (e.g. separability of dynamics). All value function approximations and control policy simulations were carried out on an Intel Core i7 Q820 CPU clocked at 1.73 GHz with 16GB of RAM memory, using IBM's CPLEX optimization toolkit in its default settings.

5.1. *Example 1: Reach-avoid origin regulation*

Consider the reach-avoid problem of maximizing the probability that the state of a controlled linear system subject to additive Gaussian noise reaches a target set centered at the origin within $T = 5$ discrete time steps while staying in an enclosing safe set. For simplicity in presenting our results, we consider systems described by the equation

$$x_{k+1} = x_k + u_k + \omega_k \quad (18)$$

Algorithm 2 Approximate controller

for $k \in \{0, \dots, T-1\}$ **do**
 Measure the system state x_k .
 Compute the gradient and hessian functions of $\int_{\mathcal{X}} \tilde{V}_{k+1}^*(y) Q(dy|x_k, u)$ with respect to u .
 Solve the optimization problem in (17) using second-order methods to obtain $\tilde{\mu}^*(x_k)$.
 Apply the calculated control input to the system.
end for

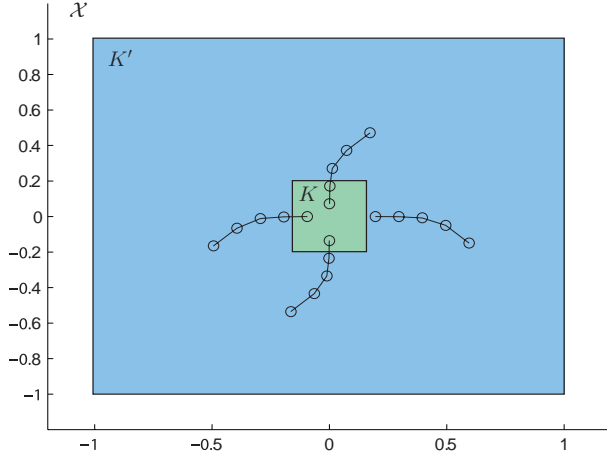


Figure 3: Example 1 - Regulation

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
M_k	100	500	1000
N_k	3960	19960	39960
ε_k	0.05	0.05	0.05
$1 - \beta_k$	0.99	0.99	0.99
$\ \tilde{V}_0 - V_{\text{ADP}}\ $	0.0692	0.1038	0.2241
Constr. time (sec)	4	85	450
LP time (sec)	2	50	520
Memory	3.2MB	80MB	320MB

Table 1: Example 1 - Parameters and properties of the constructed value function approximation for origin regulation problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

where for each $k \in \{0, \dots, 4\}$, $x_k \in \mathcal{X} = \mathbb{R}^n$, $u_k \in \mathcal{U} = [-0.1, 0.1]^n$ and each ω_k is distributed as a Gaussian random variable $\omega_k \sim \mathcal{N}(\mathbf{0}_{n \times 1}, \Sigma)$ with diagonal covariance matrix. We consider a target set $K = [-0.1, 0.1]^n$ around the origin and a safe set $K' = [-1, 1]^n$ (see Figure 3) and approximate the value function using the method described in Proposition 5 and the steps of Algorithm 1 for system dimensions of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$. The transition kernel of (18) is a Gaussian mixture $x_{k+1} \sim \mathcal{N}(x_k + u_k, \Sigma)$ with center $x_k + u_k$ and covariance matrix Σ while the sets K and K' are hyper-rectangles by construction. We choose 100, 500 and 1000 GRBF elements for the reach-avoid problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$ (Table 1) and uniform measures supported on $\bar{\mathcal{X}}$ and $[0.02, 0.095]^n$ respectively to sample the GRBF centers and variances. The violation and confidence levels for every $k \in \{0, \dots, 4\}$ are chosen to be $\varepsilon_k = 0.05$, $1 - \beta_k = 0.99$ and the measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ required to generate samples from $\bar{\mathcal{X}} \times \mathcal{U}$ is chosen to be uniform. Since there is no reason to favor some states more than others, we also choose ν to be uniform, supported on $\bar{\mathcal{X}}$. We then follow the steps of Algorithm 1 to obtain a sequence of approximate value functions $\{\tilde{V}_k\}_{k=0}^4$. To estimate the performance of the approximation, we sample 100 initial conditions x_0 uniformly from $\bar{\mathcal{X}}$ and for each one generate 100 different trajectory realizations, using the corresponding policy defined in (17) computed by Algorithm 2. We then count the number of trajectories that successfully complete the reach-avoid objective, i.e. reach K without leaving K' . In Table 1 we denote by $\|\tilde{V}_0 - V_{\text{ADP}}\|$ the mean absolute difference between the empirical success probability of the ADP controller, denoted by V_{ADP} , and the predicted performance \tilde{V}_0 , evaluated over the considered initial conditions. The memory and computation time reported is that of constructing and solving each LP in the recursive process.

The structure of the considered regulation problem is similar to that of an LQG control problem with state constraints. Since the system is linear, the noise Gaussian and the target and safe sets symmetric and centered around the origin, we assume that a tuned LQG controller will perform close to optimal for the reach-avoid objective. Under this assumption we directly compare the ADP and LQG controllers in an

attempt to validate the ADP method. Consider an LQG problem of the form

$$\begin{aligned} \min_{\{u_k\}_{k=0}^{T-1}} \mathbb{E}_{w_k} & \left(\sum_{k=0}^{T-1} x_k^\top Q x_k + u_k^\top R u_k \right) + x_T^\top Q x_T \\ \text{subject to} \quad & (18), \quad x_0 \in \bar{\mathcal{X}} \end{aligned} \quad (19)$$

where Q and R have been chosen to correspond to the largest ellipsoids² inscribed in K and \mathcal{U} respectively. Our choice is motivated by the level sets of the LQG cost function that need to correspond to the location and size of the target and control constraint sets, keeping the relative penalization between dimensions consistent. Whenever the resulting LQG control inputs (calculated via the Riccati difference equation) are infeasible, we project them on the feasible set \mathcal{U} . Starting from the same initial conditions as with the ADP controller, we simulate the performance of the LQG controller by generating 100 trajectories using the same noise samples as in the ADP method, counting the number that reach K without leaving K' . Since the design criteria of the LQG and reach-avoid based methods are different, our comparison is qualitative. Figure 4 shows the mean absolute difference between V_{LQG} and V_{ADP} over the initial conditions, as a function of the basis elements used to construct the approximate value functions. Each line on the graph corresponds to a problem of different $\dim(\mathcal{X} \times \mathcal{U})$ and we can clearly see the trend of increasing accuracy as the number of basis elements increases. In Table 2 we indicate the trade-off between accuracy and computational resources specifically for the 6D problem; the situation is analogous in the 4D and 8D problems. Figure 5 shows the same metric as Figure 4, this time as a function of the total number of sample pairs from $\mathcal{X} \times \mathcal{U}$ for a fixed number of basis elements. As expected (see Theorem 1), changing the total number of samples N_k has a direct effect on the violation probability ε_k (assuming constant $\beta_k = 0.01$) and consequently on the approximation quality. Again, as the number of samples increases, the accuracy increases at the expense of computational effort (Table 3).

5.2. *Example 2: Reach-avoid origin regulation with obstacles*

Consider the same origin regulation problem of Section 5.1 with the addition of obstacles placed randomly within the state space (Figure 6). The reach-avoid objective in this case is to maximize the probability that the system in (18) reaches K without leaving K' or reaching any of the obstacle sets K_α . In the reach-avoid framework, this problem is equivalent to the one in Section 5.1, using the same target set K and a different safe set $K' = \mathcal{X} \setminus \bigcup_{j=1}^J K_\alpha^j$ where each K_α^j denotes one of the hyper-rectangular obstacle sets and $J \in \mathbb{N}_+$. For a time horizon of $T = 7$, we choose the same basis numbers, basis parameters, sampling and reward measures as well as violation and confidence levels as in Section 5.1 in order to apply Algorithm 1 and compute approximate value functions for the obstacle avoidance problem (see Table 4). We simulate the performance of the ADP controller starting from 100 different initial conditions selected such that at least one obstacle blocks the direct path to the origin. For every initial condition we sample 100 different noise trajectory realizations and use the corresponding control policies computed by Algorithm 2 to compute the empirical ADP reach-avoid success probability (denoted by V_{ADP}) by counting the total number of trajectories that reach K while avoiding reaching any of the obstacles or leaving K' .

The problem of regulating (18) to the origin without passing through any obstacles is in the category of path planning and collision avoidance problems that have been studied thoroughly in the control community [39, 40, 41, 42] in the noiseless case. We adopt the formulation in [40] and formulate the problem as a mixed integer quadratic program (MiQP) using mixed logic dynamics (MLD) [43] that can be solved to optimality using standard branch and bound techniques. In order to take noise into account, we truncate the density function of the random variables ω_k at 95% of their total mass and enlarge each obstacle set K_α by the maximum value of the truncated ω_k in each dimension (robust worst-case approach). An alternative approach to handle process noise would be to include the bounded uncertainty implicitly in the control problem formulation, see [44], or employ randomized techniques and solve the problem using sampling.

²Computed via convex optimization [38, Ch. 8]

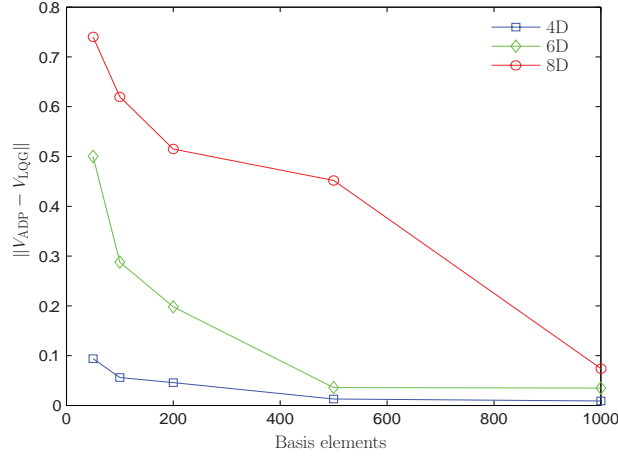


Figure 4: Example 1 - Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of approximation basis elements for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

M_k	$\ V_{\text{ADP}} - V_{\text{LQG}}\ $	Constr. time (sec)	LP time (sec)	Memory
50	0.504	1.2	0.143	784KB
100	0.288	4	2.324	3.2MB
200	0.052	15	3.41	12.8MB
500	0.036	85	50	80MB
1000	0.035	450	850	320MB

Table 2: Example 1 - Accuracy and computational requirements as a function of basis elements for $\dim(\mathcal{X} \times \mathcal{U}) = 6$.

Starting from the same initial conditions as in the ADP approach, we simulate the performance of the MiQP based control policy by using 100 noise trajectory realizations (same as in the ADP controller), implementing the policy in receding horizon. The empirical success probability of trajectories that reach K while avoiding reaching any of the obstacles or leaving K' is denoted by V_{MiQP} . Since the objectives of the MiQP and ADP formulations are different (even though both methods exploit knowledge about the target and the avoid sets) our comparison is again qualitative. The mean differences $\|\tilde{V}_0 - V_{\text{ADP}}\|$ and $\|V_{\text{ADP}} - V_{\text{MiQP}}\|$ reported in Tables 4 and 5 are computed by averaging the corresponding empirical reach-avoid success probabilities over the initial conditions. Figure 7 and Table 5 indicate a similar trade-off between accuracy and complexity as observed in Section 5.1.

5.3. Example 3: Race car cornering

Consider the problem of driving a race car through a tight corner in the presence of static obstacles, illustrated in Figure 8. As part of the ORCA project of the Automatic Control Lab (see <http://control.ee.ethz.ch/~racing/>), a six state variable nonlinear model with two control inputs has been identified to describe the movement of 1:43 scale race cars. The model derivation is discussed in [45] and is based on a unicycle approximation with parameters identified on the experimental platform of the ORCA project using model cars manufactured by Kyosho. We denote the state space by $\mathcal{X} \subset \mathbb{R}^6$, the control space by $\mathcal{U} \subset \mathbb{R}^2$ and the identified dynamics by a function $f : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{X}$. The first two elements of each state $x \in \mathcal{X}$ correspond to spatial dimensions, the third to orientation, the fourth and fifth to body fixed longitudinal and lateral velocities and the sixth to angular velocity. The two control inputs $u \in \mathcal{U}$ are the throttle duty cycle and the steering angle.

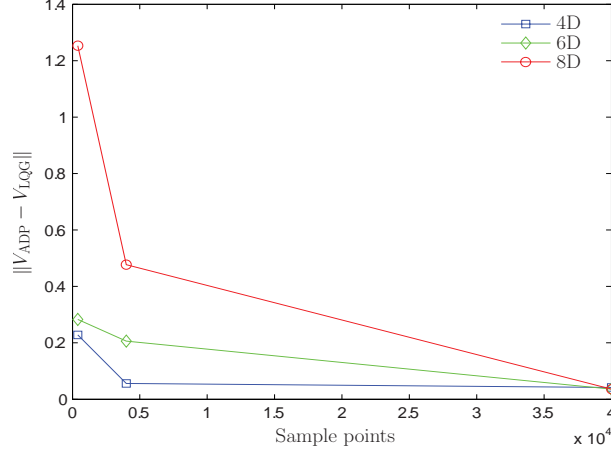


Figure 5: Example 1 - Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of samples for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

Samples	$\ V_{\text{ADP}} - V_{\text{LQG}}\ $	Constr. time (sec)	LP time (sec)	Memory
400	0.283	2.2	3.565	1.6MB
4000	0.206	17	97	16MB
40000	0.0360	170	162	160MB

Table 3: Example 1 - Accuracy and computational requirements as a function of sample number for $\dim(\mathcal{X} \times \mathcal{U}) = 6$, with the parameters of Table 1, i.e $M_k = 500$ and $1 - \beta_k = 0.99$. The empirical violation is denoted by

As is typically observed in practice, the state predicted by the identified dynamics and the state measurements recorded on the experimental platform are different due to process and measurement noise. Analyzing the deviation between predictions and measurements, we identified a stochastic variant to the original model using additive Gaussian noise,

$$g(x, u) = f(x, u) + \omega, \quad \omega \sim \mathcal{N}(\mu, \Sigma). \quad (20)$$

The noise mean μ and diagonal covariance matrix Σ have been selected such that the probability density function of the process in (20) resembles the empirical data obtained via measurements. Figure 10 illustrates the fit for the angular velocity where $\mu_6 = -0.26$ and $\Sigma(6, 6) = 0.53$; the rest of the states are handled in the same way. The kernel of the resulting stochastic process is by construction a GRBF with a single term $\mathcal{N}(f(x, u) + \mu, \Sigma)$.

We cast the problem of driving the race car through a tight corner without reaching obstacles as a stochastic reach-avoid problem. The setup is similar to the one in Section 5.2 with the crucial difference that the system is highly nonlinear and the MiQP method used before is not applicable without suitable linearization. On the contrary, the developed ADP technique can be readily applied to this problem once the input data and design parameters of Algorithm 1 are fixed. We consider a horizon of $T = 6$ and a sampling time of 0.08 seconds. The safe region on the spatial dimensions is defined as $(K'_1 \times K'_2) \setminus A$ where $A \subset \mathbb{R}^2$ denotes the obstacle set that has to be avoided by the moving car, see Figures 8, 9. The full safe set is then defined as $K' = ((K'_1 \times K'_2) \setminus A) \times K'_3 \times K'_4 \times K'_5 \times K'_6$ where K'_3, K'_4, K'_5, K'_6 describe the physical limitations of the model car (see Table 6). Similarly, the target region for the spatial dimensions is denoted by $K_1 \times K_2$ and corresponds to the end of the turn - Figure 9. The full target set is then defined as $K = K_1 \times K_2 \times K'_3 \times K'_4 \times K'_5 \times K'_6$ which contains all states $x \in K'$ for which $(x_1, x_2) \in K_1 \times K_2$. We use a total of 2000 GRBF elements for each approximation step with centers and variances sampled according to uniform measures supported on $\tilde{\mathcal{X}}$ and the hyper-rectangle defined by the product of intervals in the

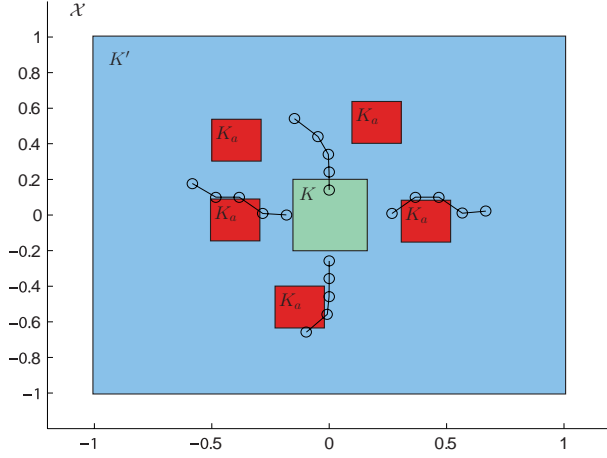


Figure 6: Example 2 - Non-convex state constraints

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
M_k	100	500	1000
N_k	3960	19960	39960
ε_k	0.05	0.05	0.05
$1 - \beta_k$	0.99	0.99	0.99
$\ \tilde{V}_0 - V_{\text{ADP}}\ $	0.095	0.118	0.191
Constr. time (sec)	4.2	130	671
LP time (sec)	3.2	80	700
Memory	3.2MB	80MB	320MB

Table 4: Example 2 - Parameters and properties of the constructed value function approximation for obstacle avoidance problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

rows of Table 6 respectively. As in Sections 5.1 and 5.2, we use a uniform state-relevance measure and a uniform sampling measure to construct each one of the finite linear programs in Algorithm 1. All violation and confidence levels are chosen to be $\varepsilon_k = 0.2$ and $1 - \beta_k = 0.99$ respectively for $k = \{0, \dots, 5\}$. Having fixed all design parameters we implement the steps of Algorithm 1 and compute a sequence of approximate value functions. To evaluate the quality of the approximations we initialized the car at two different initial states

$$x^1 = (0.33, 0.4, -0.2, 0.5, 0, 0), \quad x^2 = (0.33, 0.4, -0.2, 2, 0, 0)$$

corresponding to entering the corner at low ($x_4^1 = 0.5$ m/s) and high ($x_4^2 = 2$ m/s) longitudinal velocities. The approximate value function values $\tilde{V}_0(x^1) = 0.98$, $\tilde{V}_0(x^2) = 1$ are both high at these initial states but the associated trajectories computed via Algorithm 2 vary significantly. In the low velocity case, the car avoids the obstacle by driving above it while in the high velocity case, by driving below it; see Figure 9. Such a behavior is expected since the car model would slip if it turns aggressively at high velocities. We also computed empirical reach-avoid probabilities in simulation, $V_{\text{ADP}}(x^1) = 1$ and $V_{\text{ADP}}(x^2) = 0.99$, by sampling 100 noise trajectories from each initial state and implementing the ADP control policy of (17) using the associated value function approximation (Figure 9). The controller was tested on the ORCA setup by running 10 experiments from each initial condition and sequentially applying the computed control inputs. Implementing the whole feedback policy online would require solving problems in the form of (17) within the sampling time of 0.08 seconds. This is conceptually possible since the control space is only two dimensional but requires developing an embedded nonlinear programming solver compatible with the ORCA setup. As demonstrated by the videos in (youtube:ETHZurichIfA), the car is successfully driving through the corner, avoiding the obstacle, even when the control inputs are applied in open loop.

6. Conclusion

We developed a numerical approach to compute the value function of the stochastic reach-avoid problem using linear programming and randomized convex optimization. The method suggested is based on function approximation properties of Gaussian radial basis functions and exploits their structure to encode the transition kernel of Markov decision processes and compute integrals over the reach-avoid safe and target sets. The fact that our method relies on solving linear programs allows us to tackle reach-avoid problems with larger dimensions than state-space gridding methods. The accuracy and reliability of the approach is investigated by comparing to heuristics based on well-established methods in the control of linear systems.

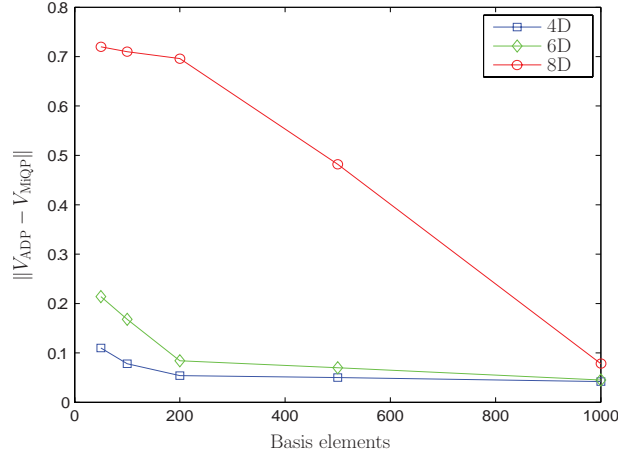


Figure 7: Example 2 - Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP and MiQP policies as a function of approximation basis elements for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

M_k	$\ V_{\text{ADP}} - V_{\text{MiQP}}\ $	Constr. time (sec)	LP time (sec)	Memory
50	0.214	1.67	0.18	784KB
100	0.168	5.59	2.66	3.2MB
200	0.084	22	4.3	12.8MB
500	0.07	130	80	80MB
1000	0.045	507	1213	320MB

Table 5: Example 2 - Accuracy and computational requirements as a function of basis elements for $\dim(\mathcal{X} \times \mathcal{U}) = 6$.

The potential of the approach is demonstrated by tackling a reach-avoid control problem for a six dimensional nonlinear system with two control inputs. Based on the results reported so far, it is worth concentrating future work to apply the method in more general MDP control problems with uncountable state space and control spaces.

We are currently focusing on the problem of systematically choosing the center and variance parameters of basis function elements, exploiting knowledge about the dynamics of the considered systems. Apart from improving initial basis selection, we are exploring adaptive methods for choosing basis parameters and state-relevance measures to increase objective performance. In terms of computational efficiency, we are developing decomposition methods for the large linear programming problems that arise in our approximation method to allow addressing reach-avoid problems of even higher dimensions. Finally, we are addressing tractable reformulations of the infinite constraints in the semi-infinite linear programs presented to avoid sampling-based methods.

Acknowledgments

The authors would like to thank Alexander Liniger from the Automatic Control Laboratory in ETH Zurich for his help in testing the algorithm on the ORCA platform.

References

- [1] E. A. Feinberg, A. Shwartz, E. Altman, Handbook of Markov decision processes: methods and applications, Kluwer Academic Publishers Boston, MA, 2002.

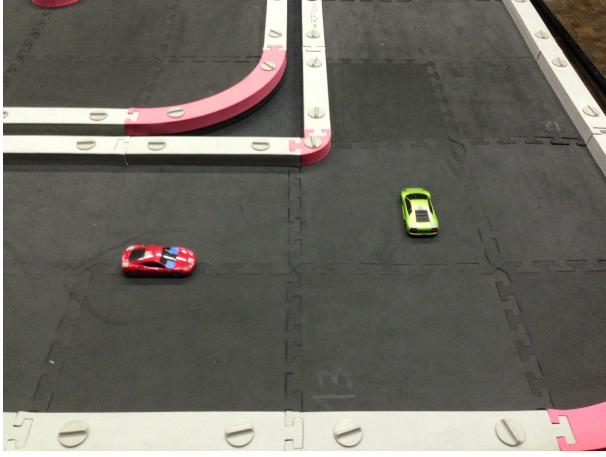


Figure 8: Example 3 - Race-car cornering example

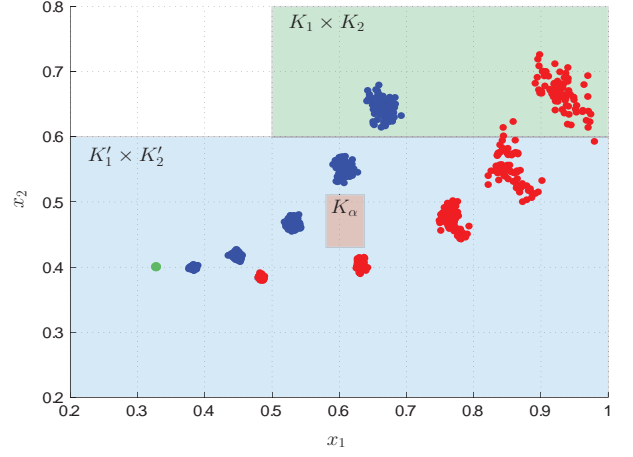


Figure 9: Example 3 - Velocity dependent cornering trajectories

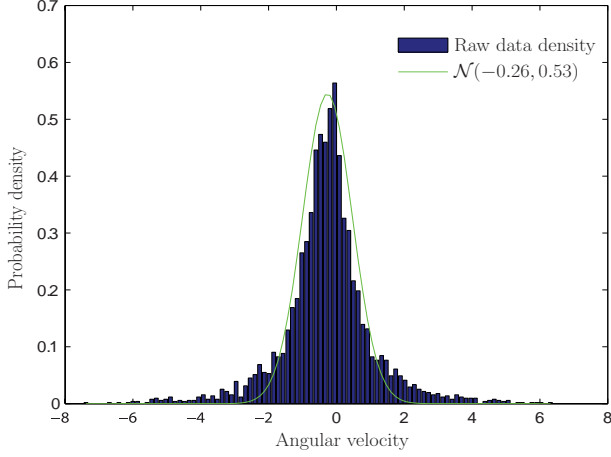


Figure 10: Example 3 - Noise fit for the angular velocity.

Safe region	min	max	Basis Variance
K'_1 (m)	0.2	1	[0.0008,0.0012]
K'_2 (m)	0.2	0.6	[0.0008,0.0012]
K'_3 (rad)	$-\pi$	π	[0.005,0.015]
K'_4 (m/s)	0.3	3.5	[0.005,0.015]
K'_5 (m/s)	-1.5	1.5	[0.005,0.015]
K'_6 (rad/s)	-8	8	[2,4]

Table 6: Example 3 - Dimension safety limits and basis variances used in ADP approximation.

- [2] M. Puterman, Markov decision processes: Discrete stochastic dynamic programming, John Wiley & Sons, Inc., 1994.
- [3] A. Abate, M. Prandini, J. Lygeros, S. Sastry, Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems, Automatica 44 (11) (2008) 2724–2734.
- [4] O. Watkins, J. Lygeros, Stochastic reachability for discrete time systems: An application to aircraft collision avoidance, in: 42nd IEEE Conference on Decision and Control (CDC), 2003. Proceedings., Vol. 5, IEEE, 2003, pp. 5314–5319.
- [5] L. Jin, H. Liu, R. Kumar, V. Ajjarapu, J. D. McCalley, N. Elia, V. Vittal, An application of reachable set analysis in power system transient stability assessment, in: Power Engineering Society General Meeting, IEEE, 2005, pp. 1715–1719.
- [6] N. Kariotoglou, D. M. Raimondo, S. J. Summers, J. Lygeros, Multi-agent autonomous surveillance: a framework based on stochastic reachability and hierarchical task allocation, Journal of dynamic systems, measurement, and control 137 (3) (2015) 031008.
- [7] T. Wood, S. Summers, J. Lygeros, et al., A stochastic reachability approach to emergency building evacuation, in: 52nd IEEE Conference on Decision and Control (CDC), 2013, IEEE, 2013, pp. 5722–5727.
- [8] A. Abate, S. Amin, M. Prandini, J. Lygeros, S. Sastry, Computational approaches to reachability analysis of stochastic hybrid systems, in: Hybrid Systems: Computation and Control, Springer, 2007, pp. 4–17.
- [9] M. Prandini, J. Hu, Stochastic reachability: Theory and numerical approximation, Stochastic hybrid systems, Automation and Control Engineering Series 24 (2006) 107–138.
- [10] H. J. Kushner, P. Dupuis, Numerical methods for stochastic control problems in continuous time, Vol. 24, Springer, 2001.
- [11] W. B. Powell, Approximate Dynamic Programming: Solving the curses of dimensionality, Vol. 703, John Wiley & Sons,

- 2007.
- [12] D. Bertsekas, Dynamic programming and optimal control, Vol. 2, Athena Scientific Belmont, MA, 2012.
 - [13] D. de Farias, B. Van Roy, The linear programming approach to approximate dynamic programming, *Operations Research* 51 (6) (2003) 850–865.
 - [14] S. Summers, J. Lygeros, Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem, *Automatica* 46 (12) (2010) 1951–1961.
 - [15] S. Summers, M. Kamgarpour, C. Tomlin, J. Lygeros, Stochastic system controller synthesis for reachability specifications encoded by random sets, *Automatica* 49 (9) (2013) 2906–2910.
 - [16] A. S. Nowak, Universally measurable strategies in zero-sum stochastic games, *The Annals of Probability* (1985) 269–287.
 - [17] L. Brown, R. Purves, et al., Measurable selections of extrema, *The annals of statistics* 1 (5) (1973) 902–912.
 - [18] D. Bertsekas, Dynamic programming and stochastic control.
 - [19] R. K. Sundaram, A first course in optimization theory, Cambridge university press, 1996.
 - [20] O. Hernández-Lerma, J. B. Lasserre, Approximation schemes for infinite linear programs, *SIAM Journal on Optimization* 8 (4) (1998) 973–988.
 - [21] E. J. Anderson, P. Nash, Linear programming in infinite-dimensional spaces: theory and applications, Wiley New York, 1987.
 - [22] D. Bertsimas, D. B. Brown, C. Caramanis, Theory and applications of robust optimization, *SIAM review* 53 (3) (2011) 464–501.
 - [23] A. Ben-Tal, A. Nemirovski, Robust optimization—methodology and applications, *Mathematical Programming* 92 (3) (2002) 453–480.
 - [24] R. Hettich, K. O. Kortanek, Semi-infinite programming: theory, methods, and applications, *SIAM review* 35 (3) (1993) 380–429.
 - [25] M. Campi, S. Garatti, The exact feasibility of randomized solutions of uncertain convex programs, *SIAM Journal on Optimization* 19 (3) (2008) 1211–1230.
 - [26] P. Mohajerin Esfahani, T. Sutter, J. Lygeros, Performance Bounds for the Scenario Approach and an Extension to a Class of Non-convex Programs, *IEEE Transactions on Automatic Control* 60 (1) (2015) 46–58.
 - [27] T. Sutter, P. M. Esfahani, J. Lygeros, Approximation of Constrained Average Cost Markov Control Processes, in: *IEEE Conference on Decision and Control*, Los Angeles, USA, 2014, pp. 6597 – 6602.
 - [28] S. M. Khansari-Zadeh, A. Billard, Learning stable nonlinear dynamical systems with gaussian mixture models, *IEEE Transactions on Robotics* 27 (5) (2011) 943–957.
 - [29] E. J. Hartman, J. D. Keeler, J. M. Kowalski, Layered neural networks with Gaussian hidden units as universal approximations, *Neural computation* 2 (2) (1990) 210–215.
 - [30] I. W. Sandberg, Gaussian radial basis functions and inner product spaces, *Circuits, Systems and Signal Processing* 20 (6) (2001) 635–642.
 - [31] J. Park, I. Sandberg, Universal approximation using radial-basis-function networks, *Neural computation* 3 (2) (1991) 246–257.
 - [32] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of control, signals and systems* 2 (4) (1989) 303–314.
 - [33] A. Bemporad, C. Filippi, F. D. Torrisi, Inner and outer approximations of polytopes using boxes, *Computational Geometry* 27 (2) (2004) 151–178.
 - [34] A. Wächter, C. Laird, F. Margot, Y. Kawajir, Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT (2009).
 - [35] G. Calafiore, M. Campi, The scenario approach to robust control design, *IEEE Transactions on Automatic Control* 51 (5) (2006) 742–753.
 - [36] A. Petretti, M. Prandini, An approximate linear programming solution to the probabilistic invariance problem for stochastic hybrid systems, in: *IEEE Conference on Decision and Control*, 2014.
 - [37] L. Deori, L. Giulioni, M. Prandini, Optimal building climate control: a solution based on nested dynamic programming and randomized optimization, in: *IEEE Conference on Decision and Control*, 2014.
 - [38] S. P. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.
 - [39] J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation* 7 (3) (1991) 278–288.
 - [40] A. Richards, J. P. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, in: *American Control Conference*, 2002. Proceedings of the 2002, Vol. 3, IEEE, 2002, pp. 1936–1941.
 - [41] J. Van Den Berg, D. Ferguson, J. Kuffner, Anytime path planning and replanning in dynamic environments, in: *Robotics and Automation*, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE, 2006, pp. 2366–2371.
 - [42] R. B. Patel, P. J. Goulart, Trajectory generation for aircraft avoidance maneuvers using online optimization, *Journal of guidance, control, and dynamics* 34 (1) (2011) 218–230.
 - [43] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica* 35 (3) (1999) 407–427.
 - [44] S. Rakovic, D. Mayne, Robust time optimal obstacle avoidance problem for constrained discrete time systems, in: *44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2005, IEEE, 2005, pp. 981–986.
 - [45] A. Liniger, A. Domahidi, M. Morari, Optimization-based autonomous racing of 1: 43 scale RC cars, *Optimal Control Applications and Methods*.